

OsmoMGW - Feature #1937

Implement way how to handle luUP on RTP endpoints

02/02/2017 01:07 PM - laforge

Status: New	Start date: 02/03/2017
Priority: Normal	Due date: 02/03/2017
Assignee: laforge	% Done: 0%
Category:	
Target version:	
Description It seems like (at least) the nano3G have a bug where they cannot handle RTP/luUP between two hNodeB for some reason. So we'll have to implement some kind of minimal RTP proxy that locally terminates the luUP protocol but passes the payload through. It should use the luUP library code to be developed in #1936	
Related issues:	
Related to Cellular Network Infrastructure - Bug #3411: 3G voice broken, CS R...	Resolved 07/23/2018
Precedes OsmoMGW - Feature #2459: remove nano3G luUP "Initialization ACK" hac...	New 02/06/2017 02/06/2017
Follows OsmoMGW - Feature #1936: Implementation of luUP SmpSDU mode	Stalled 02/02/2017

History

#1 - 02/02/2017 03:32 PM - neels

The most minimal "RTP proxy" that is apparently sufficient for nano3G operation would merely change the UP Initialization sent to a nano3G to an ACK Initialization by writing 0xe4 to the Initialization message's first byte...

This is currently done in our MGCP-GW (openbsc.git:osmo-bsc_mgcp on branch neels/nano3G), notably without detecting whether it is indeed luUP being forwarded but by crudely writing 0xe4 to the first payload forwarded in an RTP stream. Definitely needs improvement.

#2 - 02/02/2017 03:40 PM - neels

- Related to Feature #1936: Implementation of luUP SmpSDU mode added

#3 - 02/20/2017 01:55 PM - neels

neels wrote:

The most minimal "RTP proxy" that is apparently sufficient for nano3G operation would merely change the UP Initialization sent to a nano3G to an ACK Initialization by writing 0xe4 to the Initialization message's first byte...

To clarify, one side would send an UP Initialization, and the other side would send an Initialization ACK. The nano3G seems to omit the Initialization ACK even though it receives an Initialization, possibly because it assumes to be the first to have sent the Initialization and thus deserves an ACK from the other side instead of being the party that should ACK. In our case, the other side is so far the same nano3G itself, because we first echo back the RTP streams to the sender until both legs of the call are established, at which point we switch RTP from echo to forward mode. Maybe this should actually change so that the nano3Gs talk to each other directly concerning luUP Initialization and ACK, in which case a proxy might be necessary to change the second Initialization to an ACK as indicated above.

#4 - 05/24/2017 02:06 PM - neels

neels wrote:

Maybe this should actually change so that the nano3Gs talk to each other directly

right now I don't see how this can work. According to our MNCC code in OsmoMSC, we first need to establish both call legs before we can BRIDGE

two calls. Establishing a call leg centrally depends on the RAB Assignment to be successful. Having that succeed centrally depends on the UP Initialization to succeed. So it would always need to be our MGW (aka MGCP-GW aka osmo-bsc_mgcp) that responds with the Initialization ACK, IUUC.

#5 - 08/17/2017 06:51 AM - laforge

- Assignee changed from Osmocom CNI Developers to sysmocom

#6 - 09/19/2017 11:21 PM - neels

- Project changed from OsmoHNBGW to OsmoMGW

shouldn't this go to OsmoMGW? Maybe assign to dexter?

#7 - 09/23/2017 04:40 AM - laforge

Hi Neels,

On Tue, Sep 19, 2017 at 11:21:45PM +0000, neels [REDMINE] wrote:

Project changed from OsmoHNBGW to OsmoMGW
shouldn't this go to OsmoMGW?

correct, it was probably created at a time when there was no OsmoMGW project in redmine, or it was a mistake.

Maybe assign to dexter?

By now that's a good idea, yes. Wasn't clear yet a the time...

#8 - 10/11/2017 12:59 AM - laforge

- Subject changed from Implement minimal RTP proxy with luUP to Implement way how to handle luUP on RTP endpoints

- Assignee changed from sysmocom to dexter

The interesting problem here is how to express luUP in MGCP.

We cannot introduce separate endpoint types, as this would mean we can never connect an luUP (3G) and plain RTP (2G) connection together. So this support will probably be at connection level, so a single RTP-bridge endpoint can have connections either in RTP or in luUP-fashion.

#9 - 11/16/2017 09:31 PM - laforge

Based on several sources, and the fact that luUP is more or less identical to luNP, the SDP for luUP should look like this:

```
m=audio 49300 RTP/AVP 97
a=rtpmap:97 VND.3GPP.IUFP/16000
```

Where 16000 is of course only applicable in case of AMR-WB. The key part is the "VND.3GPP.IUFP" part instead of "AMR".

#10 - 11/18/2017 08:26 PM - laforge

- Precedes Feature #2459: remove nano3G luUP "Initialization ACK" hack when luUP proxy is in place added

#11 - 11/18/2017 08:27 PM - laforge

- Related to deleted (Feature #1936: Implementation of luUP SMpSDU mode)

#12 - 11/18/2017 08:27 PM - laforge

- Due date set to 02/03/2017

- Start date changed from 02/02/2017 to 02/03/2017

- Follows Feature #1936: Implementation of luUP SMpSDU mode added

#13 - 12/23/2017 07:34 PM - laforge

- Assignee changed from dexter to laforge

#14 - 12/23/2017 07:35 PM - laforge

- Priority changed from Normal to Urgent

#15 - 12/24/2017 10:41 AM - laforge

My humble guess would be that if we activate the voice bearers with "RANAP_UserPlaneMode_transparent" there would be no luUP inside the RTP to begin with.

Not aware of it; I remember that I tried to tweak around various parameters when attempting to get the nano3G to run, but am not sure whether I tweaked this one in particular. Would be good to try.

#17 - 05/30/2018 02:53 PM - laforge

- Tags set to 3G

#18 - 07/23/2018 03:39 PM - neels

- Related to Bug #3411: 3G voice broken, CS RAB Assignment is not working added

#19 - 10/17/2018 10:14 AM - laforge

- Priority changed from Urgent to Normal